

ERLANGMS: Uma Plataforma em Erlang/OTP para Modernização de Sistemas Legados através de uma Abordagem Orientada a Serviços na UnB

Everton Agilar¹, Alysson Ribeiro¹, Renato Ribeiro¹

¹Universidade de Brasília – Campus Universitário Darcy Ribeiro
Caixa Postal – 70910-90 – Brasília – DF – Brasil

{evertonagilar, rcarauta, alyssonribeiro}@unb.br

Resumo. Nos últimos anos, a modernização dos sistemas legados da Universidade de Brasília (UnB) tem sido prioridade para o CPD/UnB. A Arquitetura Orientada a Serviços (SOA) surge como uma maneira de solucionar este problema, disponibilizando uma abstração de alto nível entre as aplicações e a camada de negócio ou serviço. Este artigo aborda este tema e descreve alguns resultados obtidos com o uso da plataforma ERLANGMS desenvolvido pelo CPD/UnB sob uma abordagem orientada a serviços que compreende um processo de modernização e um barramento aderente ao estilo arquitetural Representational State Transfer (REST). Mais especificamente, são discutidos os principais resultados alcançados, as quais destacam-se um barramento de serviços orientado a contrato de serviços; um kit de desenvolvimento (SDK) para criação de serviços; um processo denominado SMSOC para guiar as atividades de migração de sistemas legados; e alguns serviços de apoio como um proxy LDAP para unificar o login de usuários e um serviço de autenticação/autorização OAuth2 para os serviços em REST.

1. Introdução

Os sistemas legados correspondem às aplicações que sustentam o funcionamento negocial de uma Instituição e consolidam a maior parte das informações corporativas [Bennett 1995]. Na Universidade de Brasília (UnB), há uma gama considerável de sistemas legados desenvolvido ao longo dos últimos 30 anos pelo CPD/UnB que consistem em um arcabouço de regras de negócios que são de vital importância para o pleno funcionamento da Instituição. Entretanto, com as sucessivas revisões nas regras de negócios para mantê-los alinhados com as necessidades e a obsolescência tecnológica desses sistemas, tornaram-se rígidos e inflexíveis, a ponto de serem de difícil manutenção e evolução.

De forma geral, os sistemas da UnB dividem-se em três áreas de negócio: área acadêmica, administrativa e de pessoal. A maioria desses sistemas foram construídos em diferentes linguagens de programação, arquiteturas e plataformas que não conversam entre si, a não ser, por meio do banco de dados. Durante muitos anos, a linguagem de programação VB foi a predominante. Os dois sistemas mais importantes escritos em VB são o *Sistema Acadêmico (SIGRA)* e o *Sistema de Pessoal (SIPES)*, sendo os demais escritos em VB.Net, C#, PHP, ASP e Java (a plataforma atual).

Neste cenário, as tradicionais práticas de manutenção deixam de atender às organizações, que buscam formas de reduzir os custos com a manutenção, maximizar

a integração entre os sistemas, torná-los mais flexíveis às mudanças de forma para prolongar sua vida útil e facilitar a evolução desses sistemas [Bisbal et al. 1999].

Este artigo apresenta a plataforma ERLANGMS desenvolvida para apoiar a modernização de sistemas legados na UnB. Mais especificamente, o artigo discute as experiências obtidas na implantação de um barramento de serviços no CPD/UnB e a implementação de alguns serviços, as quais, destacam-se:

- Desenvolvimento de serviços utilizando o processo SMSOC
- Autenticação de usuários através um serviço proxy LDAP
- Autenticação e autorização de serviços REST com um serviço OAuth2

2. Método

Para conduzir a modernização dos sistemas legados na UnB, optou-se por experimentar com a arquitetura orientada a serviços, particularmente seguindo o *estilo arquitetural REST*, adotado em muitas Instituições por causa da facilidade para invocar serviços web a partir de qualquer sistema [Fielding 2000].

Assim, foi criada a *plataforma ERLANGMS*, que compreende: (a) um barramento de serviços desenvolvido em *Erlang/OTP* para publicar os serviços; (b) um processo de modernização denominado *Software Modernization through Service Oriented Computing (SMSOC)* para guiar os trabalhos de modernização e disponibilizar uma arquitetura de software padronizada para criação dos serviços e; (c) um kit de desenvolvimento (SDK) para desenvolver os serviços na linguagem Java ¹.

Embora *SOA* seja um tema de crescente interesse por parte dos pesquisadores e da indústria, identificou-se a necessidade prévia de condução de um mapeamento sistemático para caracterizar a modernização de sistemas legados no contexto da manutenção de software [Agilar et al. 2016]. Desse modo, muitas decisões de design da plataforma foram definidas a partir deste estudo prévio onde verificou-se que a maior parte das contribuições na literatura estão relacionados aos aspectos gerenciais da modernização de software (55,88% das publicações) e há poucos relatos de contribuições que descrevem (ou validem) técnicas ou ferramentas de modernização de software, como ERLANGMS.

3. Resultados

Nesta seção são descritos os resultados deste trabalho, A Subseção 3.1 apresenta uma visão geral da plataforma ERLANGMS, a Subseção 3.2 discute as experiências como o uso do processo SMSOC. A Subseção 3.3 descreve a solução encontrada pelo CPD/UnB para unificar o login de usuários com o uso de um proxy LDAP e por fim, a Subseção 3.4 descreve o esquema de autenticação e autorização proposto para os serviços REST.

3.1. Plataforma ERLANGMS

A plataforma ERLANGMS é constituído por um barramento de serviços (*Enterprise Service Bus*— ESB) multiplataforma orientado a contratos de serviços idealizado para servir de elo entre os sistemas da Universidade e a camada de serviço (tipicamente implementada usando a linguagem Java). De acordo com [Haupt et al. 2014], um barramento permite unificar o acesso aos serviços através de uma camada intermediadora entre componentes de software (denominados serviços) e as aplicações que consomem estes serviços.

¹ Está em desenvolvimento o SDK .Net para implementação de serviços nas linguagens C# e VB.Net.

A implementação de um novo barramento (em vez da adoção de um barramento existente), possibilitou uma melhor compreensão do estilo arquitetural REST e o domínio de alguns elementos chave propostos no ERLANGMS, como a estrutura de eventos e os recursos de tolerância a falha.

A arquitetura segue o conceito de *Service Oriented Computing* (SOC), um paradigma que promove a composição de serviços *em uma rede de serviços* fracamente acoplados, com o objetivo de criar processos de negócio dinâmicos e flexíveis através da interconexão de sistemas computacionais [Haupt et al. 2014]. Dessa forma, o barramento suporta a mediação, roteamento, transformação de dados e a orquestração dos serviços. Para isso, adotou-se o estilo arquitetural REST e o formato JSON para o envio e recebimento das mensagens do cliente. Essa restrição de design teve o objetivo de facilitar a implementação do barramento e mantê-lo simples.

O esquema de comunicação da arquitetura ocorre por meio de duas vias distintas, como ilustra a Figura 1: Na primeira via, existe a comunicação do cliente para consumir algum serviço no barramento. Essa comunicação é via uma interface REST, razão pela qual o cliente (que pode ser qualquer sistema, independente da sua linguagem de programação ou plataforma) precisa suportar chamadas de serviços em REST. Na segunda via, tem a comunicação do barramento com o serviço, que está implementado em alguma linguagem de programação (Erlang, Java, etc.). Essa comunicação dá-se via sistema de mensageria disponível em Erlang que possibilita uma comunicação assíncrona com várias linguagens de programação de forma muito rápida por trafegar os dados no formato binário e com baixa latência na rede [Armstrong 2013].

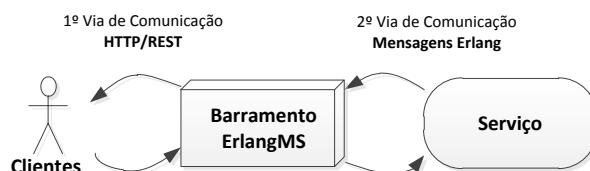


Figura 1. Roteamento das mensagens na plataforma ERLANGMS.

3.2. Desenvolvimento de serviços utilizando o processo SMSOC

A introdução de um processo surgiu devido a necessidade de documentar um processo de modernização para auxiliar os trabalhos de modernização. O processo SMSOC é aderente à arquitetura SOA e foi validado como resultado de um estudo de caso conduzido em uma disciplina de Pós-Graduação do Mestrado Acadêmico em Informática da UnB, através do qual foi modernizado o Sistema de Estudo SocioEconômico (SAE) que faz a gestão do processo de avaliação socioeconômica dos estudantes da UnB.

Para guiar as atividades de modernização, o processo divide-se em 4 fluxos de trabalho com atividades gerenciais e técnicas. Cada atividade apresenta um conjunto de diretrizes para a migração dos sistemas legados para um ambiente distribuído. O SMSOC documenta também a arquitetura e o design para implementação dos serviços e adota uma abordagem de desenvolvimento Domain-Driven Design (DDD), introduzido no livro

Domain-Driven Design: Tackling Complexity in the Heart of Software, para o desenvolvimento de sistemas complexos centrado no domínio do negócio e no trabalho cooperativo entre especialistas do negócio e desenvolvedores [Evans 2004].

Há vários benefícios observados com o SMSOC, por exemplo, por ser um processo enxuto para as necessidades da UnB, possibilita a sua execução até mesmo em projetos ou equipes muito pequenas. Além disso o processo apresenta diretrizes e boas práticas recomendadas além de conter um design padronizado da arquitetura dos serviços, o que facilita o aprendizado por novos membros na equipe de desenvolvimento e serve como uma excelente documentação. Em Dezembro de 2016, após três semanas de treinamento, a equipe de desenvolvedores da área de sistemas administrativos iniciou o desenvolvimento de vários serviços solicitados pela UnB. Em contraposição aos benefícios, identificou-se certa resistência por desenvolvedores com mais experiência na arquitetura antiga, em virtude dos sistemas legados trabalharem com a abordagem *TransactionScript* enquanto que ERLANGMS adota a abordagem DDD. Uma comparação destas duas abordagens pode ser encontrada em [Agilar 2016].

3.3. Autenticação de usuários através um serviço proxy LDAP

A UnB tem buscado desde 2013 uma forma de unificar o acesso dos usuários aos sistemas de informação e a rede corporativa, uma vez que existem muitos silos de dados de usuários que dificultam a centralização dessas informações.

De modo a suprir esta demanda e subsidiar a concretização do acesso unificado bem como o compartilhamento de informações de usuários com as aplicações (até mesmo com sistemas não desenvolvidos pelo CPD/UnB), foi implementado um serviço proxy LDAP no barramento de serviços. O LDAP ou Lightweight Directory Access Protocol é um protocolo padrão comumente utilizado pelas aplicações para autenticar os usuários, sendo um dos protocolos de autenticação mais utilizados no ambiente da UnB.

Desde que este serviço foi colocado em produção para autenticar alguns sistemas na UnB (como é o caso do Redmine e muito em breve o SEI 3.0), o proxy LDAP tem sido visto pela Direção do CPD/UnB como uma ferramenta com alto potencial para integrar não somente os sistemas da Universidade mas também a rede corporativa (Webmail e UnB Wireless).

Note que a maior vantagem de se utilizar um serviço que entende LDAP em vez de uma instância OpenLDAP convencional por exemplo, é permitir que base de usuários possa estar em um banco de dados relacional onde tipicamente os sistemas da própria Instituição já tem acesso direto. Os demais sistemas (internos ou não), sites institucionais e gerenciadores de conteúdo (Joomla, Wordpress, etc.) poderiam consultar e autenticar os usuários através do serviço proxy LDAP exatamente da mesma forma que faria se fosse um servidor OpenLDAP.

3.4. Autenticação e autorização de serviços REST com um serviço OAuth2

Com o desenvolvimento de sistemas com uma abordagem SOA, torna-se importante controlar o acesso e a autorização das chamadas de serviços REST. Nesse sentido, foi desenvolvido suporte nativo para autenticação OAuth2 no barramento de serviços.

O OAuth (*Open Authorization Protocol*) é um protocolo de autorização que permite as aplicações possam obter acesso limitado a um serviço através de trocas de men-

sagens e *tokens* de acesso. Este protocolo ainda está em desenvolvimento sendo que um estudo prévio foi conduzido para escolher o protocolo a ser adotado na solução ER-LANGMS [de Sousa Ribeiro and Canedo 2016].

Resumidamente, o suporte da autenticação OAuth2 foi dividido em duas partes no barramento de serviços. Na primeira parte realizou-se a implementação da geração do token que não é uma operação trivial. A segunda parte, que representa o desafio atual, é fazer a integração com o Sistema de Controle de Acesso (SCA) de maneira agnóstica, uma vez que o barramento pode ser utilizado por qualquer organização, sendo portanto, preferível não acoplar com os sistemas internos da UnB.

4. Conclusões

Uma das vantagens que se observou na abordagem proposta foi permitir que tanto os sistemas novos quanto os sistemas legados possam coexistir, invocando os mesmos serviços e maximizando o compartilhamento das regras de negócio.

Referências

- Agilar, E. (2016). *Uma Abordagem Orientada a Serviços para a Modernização de Sistemas Legados*. PhD thesis, Universidade de Brasília.
- Agilar, E., de Almeida, R. B., and Canedo, E. D. (2016). A systematic mapping study on legacy system modernization. pages 1–6.
- Armstrong, J. (2013). *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf.
- Bennett, K. (1995). Legacy systems: coping with success. *Software, IEEE*, 12(1):19–23.
- Bisbal, J., Lawless, D., Wu, B., and Grimson, J. (1999). Legacy information systems: Issues and directions. *IEEE software*, 1(5):103–111.
- de Sousa Ribeiro, A. and Canedo, E. D. (2016). Solutions analysis of authentication and authorization for service oriented architectures. In *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- Evans, E. (2004). *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine.
- Haupt, F., Karastoyanova, D., Leymann, F., and Schroth, B. (2014). A model-driven approach for rest compliant services. In *Web Services (ICWS), 2014 IEEE International Conference on*, pages 129–136. IEEE.